RESEARCH ARTICLE OPEN ACCESS

Implementation of Hierarchical Clusters Using the CURE Algorithm

Maghrib Abidalreda Maky Alrammahi [1], Kassem Al Attabi [2], Dhurgham Ali Mohammed [3]

1(ITRDC, University of Kufa, Najaf, Iraq
maghrib.alramahi@uokufa.edu.iq)

2(Department of computer technical Eng., College of technical engineering Islamic University Najaf, Iraq
kassem.alattabi@iunajaf.edu.iq)

3(Department of computer science, Faculty of education for girls University of Kufa Najaf, Iraq
dhurghama.alhasani@uokufa.edu.iq)

Abstract:

Data mining is one of the important concepts in the discovery of knowledge, as it works to explore a huge set of information and data that is structured or unstructured and begins with several operations, the first of which is analysis and through which it extracts meaning for the data and finally, depending on the meaning, it is possible to predict future behaviours and take the right decision for any system. Where the work begins on the databases and discovers the invisible or incomprehensible patterns and discovers the most important information that is capable of making correct predictions.

Data mining consists of four important techniques and is classified into regression and depends on prediction, descriptive (Association rules), classification, and finally clustering, and each of these techniques contains a set of algorithms. In this research paper, we will focus on the last type, which is the clustering and in particular the CURE algorithm, where this algorithm works to collect the data section by adopting a hierarchical approach, which helps to obtain the best balance and high quality in the assembly.

Keywords — **Hierarchical Clustering: CURE Algorithm; Clusters ;Data Mining.**

I. INTRODUCTION

Data mining works to discover information from big data and through algorithms, classified data or accurate results and successful compilation are obtained a successful grouping, here we need two main goals: the first is to discover and determine the similarities between data points to make a grouping or classification, and the second goal is the similar points of the data with the rest of the points that differ from the exploratory aspect from those points. There are many challenges represented by the different features of the data, some of which are continuous, others are non-continuous, and some are limited in dimensions, and there are types that are multi-dimensional in nature, therefore the different characteristics and features in the data will be addressed by clusters algorithms.

There is a very important point in clusters, which is not only to be able to distinguish between the data points only, but also to be able to distinguish comprehensively. There is another problem, which is the different and varied geometric shapes that cause a difference in the shape of the cluster. When calculating and measuring the distance of the important data points, and this is considered a problem. It should be taken care of

In the classification processes and the algorithm work, we notice the presence of data with undesirable characteristics, thus causing a difficult task in achieving and finding similarities between the data points and this leads to the creation of

inappropriate totals, and when reaching the end of the results there will be various challenges related to the interpretation of the work and results, because the work of the algorithm in The end is to obtain comprehensible outputs that are commensurate with the algorithm's criteria and the ability to properly address the problem in the end.
There are several types of aggregation methods that each method contains a set of algorithms capable of solving one or more problems that occur during execution, for example, problems in the formation of the shape of the borders or the ability to expand or grouping and noise and also the problem of dimensions, where these methods are divided into six types the main specialties in machine learning and statistical aggregation that implement the methodology are: (Connectivity-based Clustering Hierarchical clustering),(Centroids-based Clustering -Partitioning methods) , (Model-based Clustering), (Density-based Clustering) and finally (Grid-based Clustering). Figure 1 below shows in detail the above types of methods, and each method contains a set of algorithms [1].

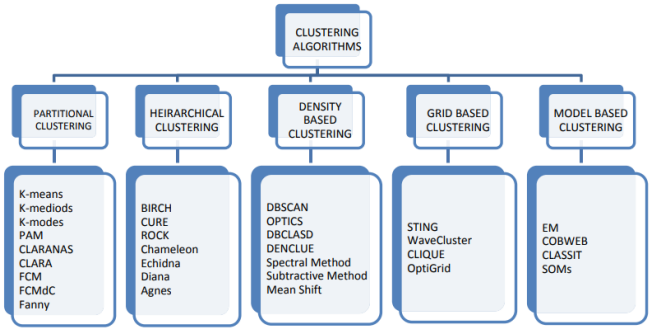

Figure 1 . Types of Clustering Algorithms[1]

- (Connectivity-based Clustering-Hierarchical clustering) : Its working principle is hierarchical groups at the beginning of the approach. For this method, a hierarchy is predetermined based on the top-down approach, and then an analysis of data objects is performed based on this pre-infiltration, and then work begins in the implementation stage with a bottom-up approach to creating clusters . It starts with single point cluster and continues move upward merging clusters and stops until it reaches the required number of clusters.
- (Centroids-based Clustering -Partitioning methods): The principle of its work is division. It starts with one cluster and continues dividing clusters and stops until it reaches the required number of clusters. Where it depends in the creation of clusters on the midpoints (central vector) by defining a group of points that are close to this centroid and works to create clusters. Relying on mathematical arithmetic methods in calculating the middle points with the rest points, for example using (Euclidian distance, Manhattan Distance).
- (Model-based Clustering) : The principle of its work is probability, as it depends on certain types in calculating and forming clusters on the probability distribution, for example (Binomial , Bernoulli ,Gaussian, etc). Where you create the aggregation through the distribution of data points and then start building on the potential cover that belongs to the data with a probability distribution.
- (Density-based Clustering): Its working principle is based on the density of homogeneous groups of random shapes, as it depends on the maximum degree of homogeneity in clusters for the same density levels. Its high ability to deal with noisy data and also with outliers.
- (Grid-based Clustering): The principle of its work is used with networks (grid), where data and information are collected statistically from the database and does not depend on direct data, as the algorithms depend mainly on the size of the network and not on the real data sizes.

II. OBJECTIVE OF RESEARCH

The Cure algorithm(dependent on type of hierarchical clustering) is considered one of the most important clustering algorithms so that it is very efficient in results and effective in comparison with the rest of the algorithms. Cure algorithm is efficient in terms of large databases and the best methods for handling outliers, which are values that are far from the generated aggregates and also efficient in their work in irregular shapes (Spherical) and the wide difference in size, which gives the best results.

III. CLUSTERING PROCEDURE

In the beginning, we need to enter a sample of the data and then go through the typical analysis stages, which consist of four steps, with the adoption of the feedback path, and finally, the final output is obtained, which is knowledge. Figure 2 shows the stages and steps[2].

- (Feature selection or extraction): In this step, some important features are identified or new features are extracted from the original text and used to create clustering in applications, as they help reduce workload and also simplify the upcoming design and are used to distinguish patterns in clusters or against noise.
- (Clustering algorithm design or selection): In this step, the algorithms are identified and used depending on the problem and the proximity scale to group the clusters according to the similarity to each other. Where at this stage the general design of the algorithm is obtained depending on the previous methods of the clusters, is the method used (hierarchical, Partitioning and other methods).
- (Clusters validation): In this step, the results are verified and the evaluation stage is also examined. This stage is very important to determine whether the results are effective and reliable and can be provided to the user. It measures the degree of confidence in the results of the assembly (clusters) extracted from the implementation of the algorithm in the previous stage.
- (Results interpretation): In this step, the user is prepared for the end goal and provides him with a useful vision based on the original data and also to ensure the high reliability of the knowledge extracted.

Note in the figure below the presence of feedback paths between the steps and these paths represent a series of simulations and also repetition is very important to obtain the final model.

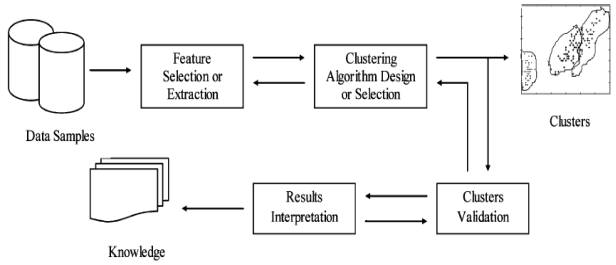

Figure 2 . Clustering procedure [2]

IV. HIERARCHICAL CLUSTERING(HC)

This type of algorithm that adopts the hierarchical method of clusters, where the results are depicted by means of a binary tree or tooth schema, represented by a group of nodes starting from the root node of the schematic tooth schematic, and each terminal node as a data object and called (child node) or intermediate nodes and these nodes are described near Organisms together, the height of the dendrogram is the distance between each pair or objects.

It is possible to obtain the final outputs in the final clusters in the dendrogram by dividing the diagram into multiple levels, this division is very important in the real hierarchical relationships in which the data is represented, for example, the evolutionary research of organizations that contain different types. The classification (HC) is based on methods of partitioning or clustering, depending on the clusters is the cumulative aggregation and each cluster contains one element(object).

Then the fusion process begins through a series of processes for all objects in the same group in an opposite way, the mitotic assembly continues. At the beginning of the process, all the data belongs to one group and through the division in succession,

individual groups or clusters will be created. There are cases of some cluster containing objects, so they are dealt with through possible divisions into two sub-groups, but this is considered costly in the implementation and computation stage.

V. IMPLEMENTATION

A. CURE ALGORITHM

This algorithm is very flexible in dealing with outliers, non-spherical shapes, and shapes of different sizes. Where the Cure algorithm works to collect a set of points that are created by selecting scattered points (representative points) from the clusters and approximately to the direction of the center of the clusters at a degree of 20 percent and the reason why the Cure algorithm adapts to non-spherical shapes is because there is more than one representative point for each group. Thus, it is able to deal with large databases and large sizes.

This algorithm uses a set of random samples and partitioning. The random samples taken from the data are partitioned, and each partition is also partially partitioned, sub-clusters then begin to aggregate to create desirable groups.

B. Steps OF algorithm

Suppose we have a data set containing 100 points (object).

Step 1: Determine the number of clusters required for the purpose of the stop condition of the algorithm, for example (K=2), where the value k represents the number of final clusters.

Step 2: Determine and choose the number of representative points (scatter points), for example(C=4), four points will be chosen to create all the clusters in the data set.

clarification:

In the second step, 4 representative points were identified and we have 100 points in the data set, thus 25 clusters will be created.

Step 3: Move the representative points by 20 % toward the center of the cluster.

Step 4: Calculate the variance (mean) centrality in each cluster through the following law:

$$mean = \frac{sumSQ}{N} - \left[\frac{sum_i}{N}\right]^2$$

where

N: represents the number of representative points for each cluster.

Sum: represents the sum of the values for each point of the x and y coordinates.

SumSQ: represents the sum of the squared values of each point for the x and y coordinates.

Step 5: rescan all points in the data set and find new points, for example (p= new points).

Step 6: Calculate the minimum distance (dmin) between the new points (p) with all the representative points in all the clusters and finally the new points of the clusters with the shortest distance between them will be merged.

Euclidean law is used to calculate the minimum distance, which is

$$d_{min} = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$$

Step 7: Merging clusters will stop until we reach the goal (k=2)

clarification :

Steps (4,5,6) will be repeated continuously in the work of the algorithm until we reach the stopping condition as shown in the first step, which is (k=2).

Important note:

There are two things that must be kept in mind in the algorithm while working and they are

First: Avoiding merging means that it is necessary to take into account the degree of separation between the selected points in all the clusters, so that they should not be too close together or too far apart, but rather far apart for a certain distance, because after determining the first shape of the clusters at the beginning of the work by the representative points, the second shape of the clusters must be far for the first so that there is no overlap between them in the other repetitions.

Second: Fraction distance means that if there is convergence, the fraction will occur and be a reason for attracting new points (p) therefore, if the convergence occurs, it will be a case of merging between two nodes.

C. Methodology

Through the flowchart below, the steps of the Cure algorithm will be clarified, which initially shows entering the dataset and then determining the initial

values for the number of clusters and continuing to calculate the variance and choosing the lowest path as shown in the figure below and finally reaching the basic condition in the case of continuation or stopping.

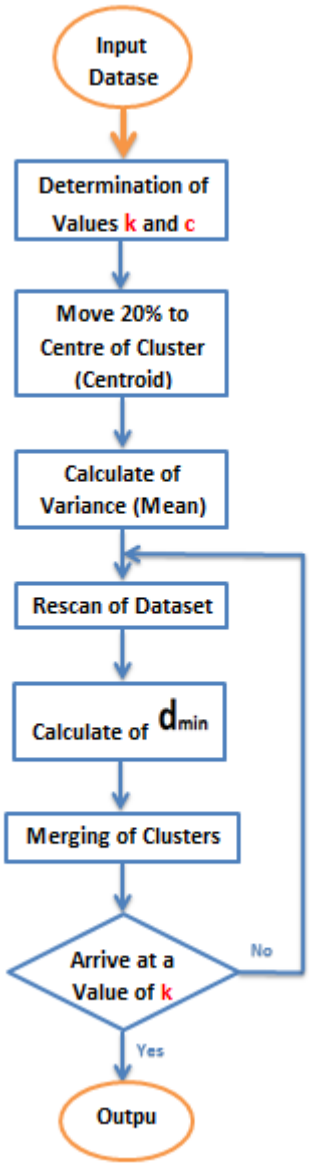

Figure 3 : Flowchart of CURE Algorithm

D. CONFIGURATION

The following code (Code 1), through the use of the Python program, will write the special codes for the Cure algorithm, in the beginning, a sample of the dataset is selected and then it is collected using the hierarchical clustering method. The values of the cluster (K) need to be determined using the dendrogram and then the representative points (N) need to be determined and moved towards the centrality of the clusters with a value (P%) and then we recheck all the dataset, Then we calculate the closest points by calculating the least path between point X and cluster C.

```
import numpy as np
import sys
import datetime

def comparator(x, y):
    if x[0] < y[0]:
        return -1
    if x[0] == y[0]:
        if x[1] < y[1]:
            return -1
        if x[1] > y[1]:
            return 1
        return 0
    return 1
samplefile = "sample_data.txt"
fullfile = "full_data.txt"

ratio = 0.2
list = []
n = 0
sum = 0
k = 3
start = datetime.datetime.now()
```

```
#hierachy clustering
with open(samplefile, "r") as file:
    lines = file.readlines()
    for line in lines:
        value = line.strip().split(",")
        list.append([[float(value[0]),float(value[1])], 1,
 [[float(value[0]),float(value[1])]]])
        n += 1

num = 1
while (n != 3):
    min_coordinate = None
    min_distace = sys.maxsize
    num += 1
    for i in range(len(list)):
        cluster1 = list[i]
        centroid1 = np.array(cluster1[0]) / cluster1[1]
        for j in range(len(list)):
            if j > i:
                cluster2 = list[j]
                centroid2 = np.array(cluster2[0]) / cluste
r2[1]

                distance = (centroid2 - centroid1) ** 2
                if np.sum(distance) < min_distace:
                    min_coordinate = [i, j]
                    min_distace = np.sum(distance)

    cor1 = list[min_coordinate[0]]
    cor2 = list[min_coordinate[1]]
```

```
    new_cluster = [np.ndarray.tolist(np.array(cor1[0]) + n
p.array(cor2[0])), cor1[1] + cor2[1], cor1[2] + cor2[2]]
    (list.remove(cor1))
    (list.remove(cor2))
    (list.append(new_cluster))

    n -= 1

centroid = []
# find representatives and calculate centroids
def reduce(function, iterable, initializer=None):
    it = iter(iterable)
    if initializer is None:
        value = next(it)
    else:
        value = initializer
    for element in it:
        value = function(value, element)
    return value
id = 0
re = []
for cluster in list:

    sortedcluster = sorted(cluster[2], cmp = comparator)
    repre = [sortedcluster[0]]
    iter = 0
    while (iter < n):
        iter += 1
        candi = None
        min_distace = sys.maxint
        if len(repre) == 1:
            center = np.array(repre)
        else:
            center = reduce(np.add, np.array(repre)) / len
(repre)
        for coor in sortedcluster:
            if coor not in repre:
                dis = np.sum((center -
 np.array(coor)) ** 2)
                if dis < min_distace:
                    min_distace = dis
                    candi = coor
        repre.append(candi)
    clustercentroid = reduce(np.add, np.array(cluster[2]))
 / len(cluster[2])
    centroid.append(clustercentroid)
    l = []
    for representative in repre:
        representative = np.ndarray.tolist(np.array(repres
entative) + ratio * (clustercentroid -
 np.array(representative)))
        l.append(representative)
    re.append(l)

# CURE
with open(fullfile, "r") as file:
    lines = file.readlines()
    points =  map(lambda line: [float(x) for x in line.str
ip().split(",")],lines)
    for point in points:
        min_dis = sys.maxint
        indexcluster = 0
        for i in range(len(re)):
            cluster = re[i]
            # print "no:", i
            for r in range(len(cluster)):
                representative = cluster[r]
                dis = np.sum((np.array(representative) -
 np.array(point)) ** 2)
                if dis < min_dis:
                    min_dis = dis
                    indexcluster = i

        list[i][2].append(point)


for x in range(len(list)):
    print (list[x][2])
```

Code 1: configurations.

VI. EXPERIMENTAL RESULT

It works on an initial set to find a specific sample and then the sample is pulled from the disk and after the sample is completed, it will be collected using hierarchical clusters. The distance to the clusters is determined by calculating the two nearest points, finally the clusters (k) will be found through the correct cutting of the dendrogram.

We have a set of points, which are represented by (X,Y) coordinates, and the stopping condition for creating the final clusters is 2, the algorithm starts with the implementations of its steps, which is to determine the representative points (the spread points), move towards the centrality of the clusters, calculate their variance (mean), and then calculate the shortest path between all The points are in the totals of the clusters and the rest of the points are combined and continue until you reach only two clusters.

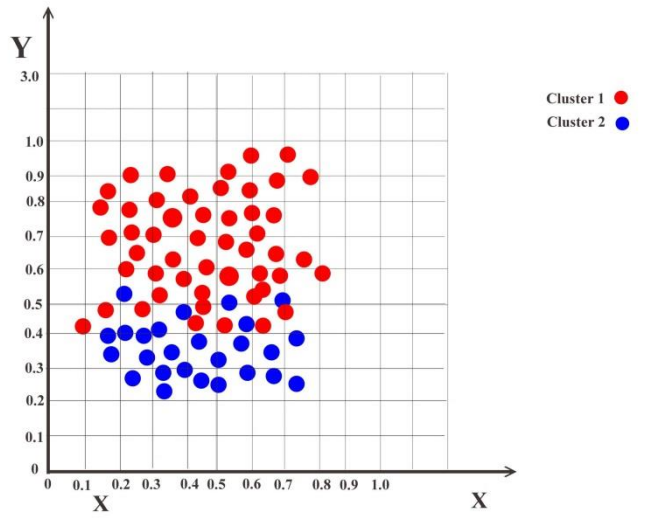

Figure 4 : Result of Hierarchical

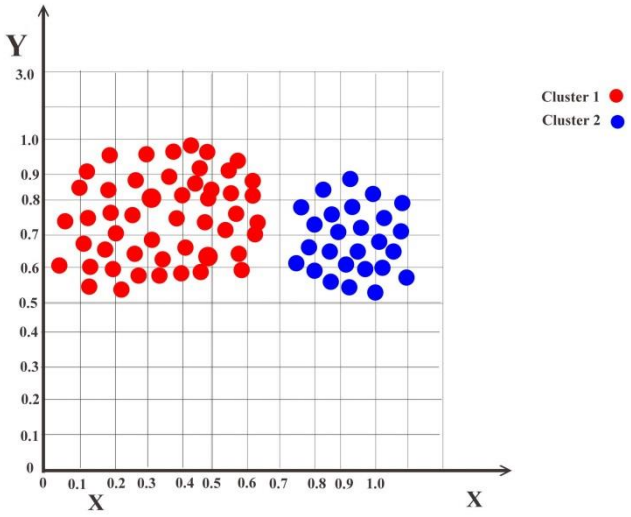

Figure 5 : Result of Reduce

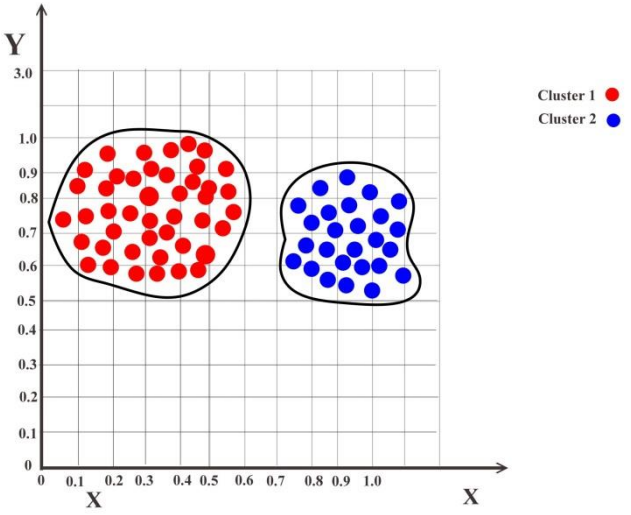

Figure 6 : Result of Final

VII. CONCLUSION AND FUTURE WORKS

Through this paper, the general vision of the types of methods in clusters was clarified and one of these types are hierarchical clusters and the steps from the beginning of data entry to the stage of analysis and design in general were explained in detail. A specific algorithm was chosen, which is the Cure algorithm, which has many benefits in analysing and obtaining accurate outputs, which deals with huge data volumes with non-spherical and unorganized shapes and also good handling of points that know outliers. This paper provided a full explanation of all the steps of the algorithm with an explanation of the flowchart and the last of its practical implementation and obtaining pictures of the results.

The future work is trying to implement the CURE algorithm with a huge number of points, choosing a large number of representative points and then examining the results and making sure of the accuracy of the outputs. Also, it will be confirmed that there are many overlaps and their impact on the rest of the shapes of the clusters during implementation in case the points increase dramatically.

REFERENCES

1. Ghosal, A., Nandy, A., Das, A. K., Goswami, S., & Panday, M. (2020). A short review on different clustering techniques and their applications. *Emerging technology in modelling and graphics*, 69-83.
2. Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, *16*(3), 645-678.
3. Ma, L., & Fan, S. (2017). CURE-SMOTE algorithm and hybrid algorithm for feature selection and parameter optimization based on random forests. *BMC bioinformatics*, *18*(1), 1-18.
4. Sisodia, D., Singh, L., Sisodia, S., & Saxena, K. (2012). Clustering techniques: a brief survey of different clustering algorithms. *International Journal of Latest Trends in Engineering and Technology (IJLTET)*, *1*(3), 82-87.
5. Patel, S., Sihmar, S., & Jatain, A. (2015, March). A study of hierarchical clustering algorithms. In *2015 2nd International Conference on*

Computing for Sustainable Global Development (INDIACom) (pp. 537-541). IEEE.

6. Rani[1], Y., & Rohil, H. (2013). A study of hierarchical clustering algorithm. *ter S & on Te SIT*, *2*, 113.
7. Yang, Y., Lian, B., Li, L., Chen, C., & Li, P. (2014, October). DBSCAN clustering algorithm applied to identify suspicious financial transactions. In *2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery* (pp. 60-65). IEEE.
8. Murtagh, F., & Contreras, P. (2017). Algorithms for hierarchical clustering: an overview, II. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *7*(6), e1219.
9. Reddy, C. K., & Vinzamuri, B. (2018). A survey of partitional and hierarchical clustering algorithms. In *Data clustering* (pp. 87-110). Chapman and Hall/CRC.
10. Bateni, M. H., Behnezhad, S., Derakhshan, M., Hajiaghayi, M. T., Kiveris, R., Lattanzi, S., & Mirrokni, V. (2017, December). Affinity clustering: Hierarchical clustering at scale. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (pp. 6867-6877).
11. Cohen-Addad, V., Kanade, V., Mallmann-Trenn, F., & Mathieu, C. (2019). Hierarchical clustering: Objective functions and algorithms. *Journal of the ACM (JACM)*, *66*(4), 1-42.
12. Chavent, M., Kuentz-Simonet, V., Labenne, A., & Saracco, J. (2018). ClustGeo: an R package for hierarchical clustering with spatial constraints. *Computational Statistics*, *33*(4), 1799-1822.
13. Govender, P., & Sivakumar, V. (2020). Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019). *Atmospheric Pollution Research*, *11*(1), 40-56.
14. Latifi-Pakdehi, A., & Daneshpour, N. (2021). DBHC: A DBSCAN-based hierarchical clustering algorithm. *Data & Knowledge Engineering*, 101922.
15. Dutta, A. K., Elhoseny, M., Dahiya, V., & Shankar, K. (2020). An efficient hierarchical clustering protocol for multihop Internet of vehicles communication. *Transactions on Emerging Telecommunications Technologies*, *31*(5), e3690.
16. Nasiakou, A., Alamaniotis, M., Tsoukalas, L. H., & Karagiannis, G. (2017, August). A three-stage scheme for consumers' partitioning using hierarchical clustering algorithm. In *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)* (pp. 1-6). IEEE.
17. Kimes, P. K., Liu, Y., Neil Hayes, D., & Marron, J. S. (2017). Statistical significance for hierarchical clustering. *Biometrics*, *73*(3), 811-821.
18. Ahmad, A., & Khan, S. S. (2019). Survey of state-of-the-art mixed data clustering algorithms. *Ieee Access*, *7*, 31883-31902.
19. Rappoport, N., & Shamir, R. (2018). Multi-omic and multi-view clustering algorithms: review and cancer benchmark. *Nucleic acids research*, *46*(20), 10546-10562.
20. Roux, M. (2018). A comparative study of divisive and agglomerative hierarchical clustering algorithms. *Journal of Classification*, *35*(2), 345-366.